CENTRE DE RECHERCHE SUR LES TRANSPORTS
(C.R.T.)
CENTRE FOR RESEARCH ON TRANSPORTATION

UNIVERSITÉ DE MONTRÉAL
ÉCOLE DES HAUTES ÉTUDES COMMERCIALES
ÉCOLE POLYTECHNIQUE

CENTRE DE RECHERCHE SUR LES TRANSPORTS
UNIVERSITÉ DE MONTRÉAL
C.P. 6128, SUCCURSALE A.
MONTRÉAL
CANADA H3C 3J7

TÉLÉPHONE: (514) 343-7575
TÉLEX: (21) 05-561325
EMAIL: crt@crt.umontreal.ca
TÉLÉCOPIEUR (FAX): (514) 343-7121

CRT-777

A Tabu Search Heuristic for the Vehicle Routing Problem

by

Michel Gendreau[1]
Alain Hertz[2,3]
Gilbert Laporte[1,2]

[1] Centre de recherche sur les transports Université de Montréal, C.P. 6128, succursale A, Montréal, H3C 3J7 Canada.
[2] GERAD, École des Hautes Études Commerciales, 5255 avenue Decelles, Montréal, H3T 1V6 Canada.
[3] Département d'Informatique et de Recherche opérationnelle, Université de Montréal, C.P. 6128, succursale A, Montréal, H3C 3J7 Canada .

Centre de recherche sur les transports — Publication #777
Groupe d'études et de recherche en analyse des décisions — Publication #G-91-31

June 1991

91-06351/

Dépôt légal – 2ème trimestre 1991

ABSTRACT

The purpose of this paper is to describe a new tabu search heuristic for the vehicle routing problem with capacity and route length restrictions. The algorithm starts with a set of routes containing one city each and then performs tabu moves consisting of inserting cities into different routes. This is done by means of a generalized insertion procedure previously developed by the authors. During the course of the algorithm, infeasible solutions are allowed. The algorithm ends with the application of a post-optimization procedure. Numerical tests on a set of benchmark problems indicate that the proposed algorithm outperforms previous heuristics.

Keywords: vehicle routing problem, tabu search, generalized insertion.

RÉSUMÉ

Cet article décrit un nouvel algorithme de type tabou pour la construction de tournées de véhicules avec contraintes de capacité et de longueur. L'algorithme part d'un ensemble de tournées contenant chacune une seule ville. On déplace ensuite des villes vers d'autres tournées, selon les règles employées dans les méthodes de type tabou. On utilise ici un algorithme d'insertion généralisée préalablement développé par les auteurs. Au cours de l'algorithme, on permet l'existence de solutions irréalisables. L'algorithme se termine par une phase de post-optimisation. Des tests numériques effectués sur des problèmes connus indiquent que l'algorithme proposé se compare avantageusement à d'autres méthodes.

Mots-clés: problèmes de construction de tournées de véhicules, méthodes de type tabou, insertion généralisée.

1. Introduction

The purpose of this paper is to present a new heuristic for the following version of the *Vehicle Routing Problem* (VRP). Let $G = (V, A)$ be a graph where $V = \{v_0, v_1, ..., v_n\}$ is a vertex set, and $A = \{(v_i, v_j) : i \neq j\}$ is an arc set. Vertex v_0 denotes a *depot* at which m identical vehicles are based, and the remaining vertices of V represent *cities* or *clients*. With every arc (v_i, v_j) is associated a non-negative distance c_{ij}. (For the sake of simplicity, the terms "distances", "travel times" and "travel costs" will be used interchangeably). The VRP consists of designing a set of least cost vehicle routes in such a way that

(i) every route starts and ends at the depot;

(ii) every city of $V \setminus \{v_0\}$ is visited exactly once by exactly one vehicle and

(iii) some side constraints are satisfied.

We consider the following side constraints:

(iiia) With every client is associated a non-negative demand $q_i, (q_0 = 0)$. The total demand of any vehicle route may not exceed the vehicle capacity Q.

(iiib) Every client v_i requires a service time $\delta_i, (\delta_0 = 0)$. The total length of any route (travel plus service times) may not exceed a preset bound L.

In our version of the problem, vehicles bear no fixed cost and their number is a decision variable.

The VRP lies at the heart of distribution management and has been extensively studied over the last three decades or so (see the surveys by Christofides, Mingozzi and

Toth (1979), Bodin, Golden, Assad and Ball (1983), Christofides (1985a), Laporte and Nobert (1987), and the book by Golden and Assad (1988)). The VRP is a hard combinatorial problem and to this day, only relatively small VRP instances can be solved to optimality. One of the best exact algorithms available uses dynamic programming coupled with state space relaxation (Christofides, Mingozzi and Toth, 1981). According to Christofides (1985b), this algorithm can *consistently* solve problems with up to 50 cities.

We are mostly interested here in heuristic algorithms. Extending the scheme proposed by Christofides (1985a), these algorithms can be broadly classified into four types.

Constructive algorithms gradually build vehicle routes by inserting at each step an unrouted city according to some criterion. One of the most famous methods belonging to this category is due to Clarke and Wright (1964). Starting with n back and forth routes between the depot and a city, it gradually merges them according to a savings criterion. The main drawbacks of this algorithm are that it can be quite time consuming (it runs in $O(n^2 \log n)$ time) and it has a tendency to produce circumferential routes. Computational complexity can, however, be reduced through the use of appropriate data structures (see, for example, Nelson, Nygard, Griffin and Shreve (1985) and Paessens (1988)). In addition, a number of authors have proposed algorithmic improvements and variants to the savings method in order to arrive at more acceptable solutions (Gaskell, 1967; Yellow, 1970; Paessens, 1988). Others, such as Mole and Jameson (1976), have used constructive methods that favour radial

shaped routes. Following an idea proposed by Dror and Levy (1986), some authors such as Altinkemer and Gavish (1985) and Desrochers and Verhoog (1989) have developed more sophisticated algorithms that consider the savings associated with all possible merges of two routes, by solving a weighted matching problem.

Two-phase algorithms can be subdivided into three main classes. In *cluster-first route-second* methods, clients are first assigned to vehicles and each route is then optimized individually by using any of the known Traveling Salesman Problem (TSP) algorithms. The sweep algorithm belongs to this category. Its origins can be traced back to two papers by Wren (1971) and by Wren and Holliday (1972), but the most commonly used version is due to Gillett and Miller (1974). Starting from an arbitrary city, the algorithm "sweeps" the plane by rotating a radius with one extremity at the depot, until a city is met where vehicle capacity becomes exceeded. The process is then repeated starting from the last city, until all cities have been swept. This initial assignment is then updated and routes are reoptimized. The two-phase algorithm of Christofides, Mingozzi and Toth (1979) is more sophisticated in that it considers several clustering trials governed by user controlled parameters. A refinement to this method was later proposed by Toth (1984). *Route-first cluster-second* methods are mostly derived from the work of Haimovich and Rinnooy Kan (1985). Typically, these algorithms first construct an optimal TSP tour and then break it into feasible VRP routes. Finally, a number of *iterative* procedures have been proposed. Fisher and Jaikumar (1981) use integer linear programming to execute the two phases, coupled with a Benders' decomposition technique to iterate between the

solution of a generalized assignment problem and that of a TSP. Noon, Mittenthal and Pillai (1991) use a Lagrangean relaxation method in which cities are assigned "reward values" and the constraint that each city is visited exactly once by a vehicle is relaxed. In the course of the algorithm, rewards are updated so as to move closer to a feasible solution. The process does not, however, always converge toward a feasible solution.

Incomplete optimization algorithms consist of obtaining a good solution by means of an incomplete search tree. For this, any of the known enumerative algorithms can be used (see Laporte and Nobert, 1987). Christofides, Mingozzi and Toth (1979) describe such a tree search procedure, based on a previous algorithm by Christofides (1976).

Improvement methods such as r-opt (Lin, 1965, Lin and Kernighan, 1973) and Or-opt (Or, 1976) interchange schemes can be used as post-optimizers on any individual route produced by a VRP algorithm, or they can be modified so that they can be applied to a global VRP solution, thus allowing for reassignments of clients to different routes. Harche and Raghavan (1991) use a generalized edge exchange procedure coupled with a relaxation of capacity constraints. *Tabu search* can be viewed as an improvement method. This metaheuristic was proposed by Glover (1977) – see Glover (1989, 1990) for a recent survey. Here, successive "neighbours" of a solution are examined and the best is selected. In order to avoid local optima, the objective is allowed to deteriorate, and to prevent cycling, solutions that were recently examined are forbidden and inserted in a constantly updated tabu list. We are aware of three VRP algorithms based on this approach. The first one is due

to Willard (1989). The problem is first transformed into a TSP by replication of the depot, and the search is restricted to neighbour solutions that can be reached by means of 2-opt or 3-opt interchanges, while satisfying the VRP constraints. In the second approach (Pureza and França, 1991), the search proceeds from one solution to the next by swapping clients between two routes. The third tabu algorithm was developed by Semet and Taillard (1991) for the solution of a real-life VRP containing several features, and different from the version considered in this paper. Here the basic tabu move consists of moving a city from its current route into an alternative route. In all three cases, a feasible solution is never allowed to become infeasible with respect to side constraints.

Our purpose is to propose a new tabu search procedure for the VRP. It differs from the three implementations just described in several fundamental aspects. Our results show that the proposed algorithm outperforms alternative methods on a set of benchmark problems. The remainder of this paper is organized as follows. The algorithm is presented in Section 2, and the computational results in Section 3. This is followed by the conclusion, in Section 4. We also provide, in an Appendix, the full solutions obtained by our algorithm on the test problems.

2. Algorithm

This section contains a description of the algorithm (TABUROUTE), followed by some comments. First, some notation is necessary. At any stage of the algorithm, the solution consists of a number of routes (feasible or infeasible) indexed by r. Route r is a vector of the form $R_r = (v_0, v_{i_1}, v_{i_2}, ..., v_0)$. For convenience, we write $v_i \in R_r$

if v_i is a component of R_r, and $(v_i, v_j) \in R_r$ if v_i and v_j are two consecutive vertices of R_r. With any feasible solution, we associate the objective function

$$F_1 = \sum_r \sum_{(v_i,v_j)\in R_r} c_{ij}.$$

Also, with any solution (feasible or not), we associate the objective

$$F_2 = F_1 + \alpha \sum_r \left[\left(\sum_{v_i \in R_r} q_i \right) - Q \right]^+ + \beta \sum_r \left[\left(\sum_{(v_i,v_j)\in R_r} c_{ij} + \sum_{v_i \in R_r} \delta_i \right) - L \right]^+,$$

where $[x]^+ = \max\ (0, x)$ and α, β are two positive parameters. If the solution is feasible, F_1 and F_2 coincide; otherwise, F_2 incorporates two penalty terms for excess vehicle capacity and excess route duration.

At a given iteration t, a new solution is obtained by removing a vertex v from its route R_r, and by inserting it in route $R_s (s \neq r)$. Note that route R_r may become empty through this process. In order to avoid falling back too soon on a previously encountered solution, a move is declared *tabu* for θ iterations, where θ is a parameter allowed to vary during the algorithm. Hence if at iteration t vertex v is removed from route R_r, reinserting v into R_r is forbidden until iteration $t + \theta + 1$. As is common in tabu search methods, a tabu move can be overridden if this results in a new overall best solution. Here, the best known values of F_1 and F_2 are recorded as F_1^* and F_2^*. A tabu move is allowed if it results in a feasible solution with a cost $F_1 < F_1^*$, or in an infeasible solution with a cost $F_2 < F_2^*$.

ALGORITHM TABUROUTE

Step 1. (Initialisation) Compute the h nearest neighbours of every vertex, where $h := \min(n, 10)$. Let $\tilde{m}$ be an estimate of the number of vehicles in the optimal solution, e.g. $\tilde{m} = \left\lceil \left(\sum_{i=1}^{n} q_i \right) / Q \right\rceil$. Define an initial solution consisting of n back and forth routes $R_r = (v_0, v_r, v_0), r = 1, ..., n$. If any of these routes is infeasible, no feasible solution exists; stop. Otherwise, compute F_1 and set $F_1^* := F_2^* := F_1$. Set $\alpha := \beta := 1$ and initialize the iteration count $t := 0$.

Execute Step 2 with $p := \min(n, 5)$ and $q := \min(n, 5\tilde{m})$ until no improvement in F_1^* or F_2^* has occured for $50n$ iterations.

Step 2. (Tabu Search) Set $t := t + 1$.

If t is a multiple of 10, check whether all previous 10 solutions were feasible with respect to vehicle capacity. If so, set $\alpha := \alpha/2$; if they were all infeasible, set $\alpha := 2\alpha$. Similarly, if all previous 10 solutions were feasible with respect to route length, set $\beta := \beta/2$; if they are were all infeasible, set $\beta := 2\beta$.

Consider all solutions that can be obtained by inserting one of q randomly selected vertices into another route that contains at least one of its p nearest neighbours. (Here, we use the generalized insertion procedure (GENI) described in Gendreau, Hertz and Laporte, 1990). In each case, compute F_2. If the solution is feasible, then $F_1 := F_2$. Determine, among allowed moves, the move providing the smallest value of F_2. If $F_2 < F_2^*$, set $F_2^* := F_2$. If the solution is feasible and $F_1 < F_1^*$, set $F_1^* := F_1$. If the best move consists of inserting vertex v from route R_r into another route, then the move consisting of reinserting v into R_r is declared tabu for θ iterations, where

θ is a random integer in [5,10].

Execute Step 2 with $p := \min(n, 10)$ and $q = n$ until no improvement in F_1^* or F_2^* has occured for 100 iterations.

Step 3. (Improvement) Apply to each individual route the US tour improvement procedure also described in Gendreau, Hertz and Laporte (1990). ■

Some comments are now in order. The idea of using a variable tabu list length (θ) was proposed by Taillard (1990). After extensive experiments on the application of tabu search to the quadratic assignment problem, this author concludes that the probability of obtaining a global optimum is increased in the case of a variable list length. Our implementation of random duration tabus differs from that proposed by Taillard since no tabu list is actually used. Instead, each move individually receives a random duration *tabu tag*: this limits the amount of bookkeeping required and, as a result, the speed of the algorithm is increased.

The idea of updating α and β during the algorithm could also be used in other contexts where penalty terms are added to the objective function. All too often, choosing an appropriate coefficient value is difficult, and a wrong choice can have an adverse impact on the performance of the algorithm. Here penalty coefficients are doubled if the 10 previous solutions were infeasible and halved if the 10 previous solutions were feasible. With this rule, we quickly arrive at values of α and β that produce a mix of feasible and infeasible solutions. In this type of algorithm, obtaining infeasible solutions is important since this helps moving out of local optima.

A key feature of TABUROUTE is the use of the GENI and US procedures. As

these have already been described elsewhere, we only sketch them here. GENI is a generalized insertion routine originally developed for the TSP. It is less myopic and more powerful than standard insertion procedures (see, e.g., Golden and Stewart, 1985) in that every insertion is executed simultaneously with a local reoptimization of the tour. Let $v \in R_r$ be a vertex identified for insertion in a different route and let $N_p^s(v)$ be the (non-empty) subset of the p nearest neighbors of v lying on some route $R_s (s \neq r)$. Let $v_i \in N_p^s(v)$. One possibility is to insert v between v_i and its successor v_{i+1} on route R_s. Two more sophisticated types of insertion can be defined as follows. Select $v_j \in N_p^s(v)$ and let v_{j+1} be its successor on R_s. Select $v_k \in N_p^s(v_{i+1})$ and $v_\ell \in N_p^s(v_{j+1})$. Denote by $v_{k-1}, v_{\ell-1}, v_{k+1}$ and $v_{\ell+1}$ the predecessors and successors of v_k and v_ℓ, respectively, on R_s. Then the insertion of v between v_i and v_j on route s can be accomplished in one of the two ways shown in Figure 1. US is a post-optimization procedure that removes in turn all vertices of a tour and reinserts them back in the most favourable position. The first phase of US is called "unstringing" the tour and it basically consists of executing GENI backwards; the second phase, called "stringing", is identical to GENI. Again, US has produced highly satisfactory results on the TSP, better than Or-opt, for example.

This algorithm contains several parameters and a number of variants can easily be envisaged. Several tests and a considerable amount of fine tuning were carried out in order to arrive at the current version. Without claiming that perfection has been reached, we can safely state that TABUROUTE is efficient, robust, and produces very good solutions within reasonable computing times. These will now be examined.

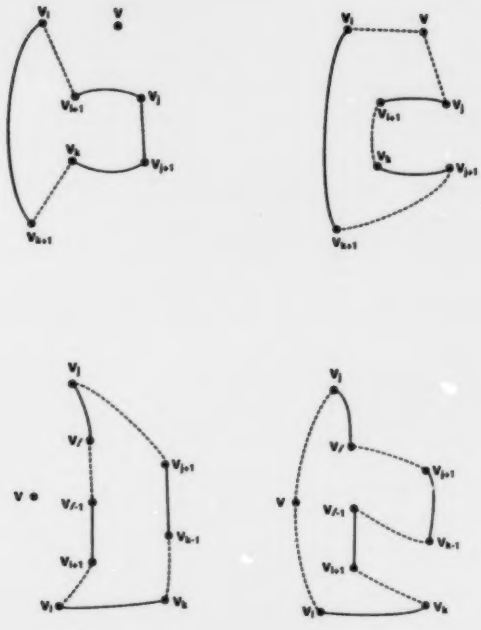

Figure 1 : Two types of insertion of v between v_i and v_j on R_s.

3. Computational results

TABUROUTE was tested on the fourteen test problems described in Christofides, Mingozzi and Toth (1979). These problems contain between 50 and 199 cities in addition to the depot. Problems 1 to 5 and 11–12 have capacity restrictions only. Problems 6 to 10 are the same as 1 to 5, except that they have a route length constraint as well; problems 13–14 are also the same as 11–12, with a route length constraint. In problems 1–10, cities are randomly generated in the plane, while in problems 11–14, they appear in clusters.

Comparisons were made with other heuristic algorithms for which computational results have already been reported for the same problems:

CW : Clarke and Wright (1964);

MJ : Mole and Jameson (1976);

AG : the Altinkemer and Gavish (1985) PSA-T algorithm;

DV : the Desrochers and Verhoog (1989) MBSA algorithm;

SWEEP : Gillett and Miller (1974);

TP : the Christofides, Mingozzi and Toth (1979) two-phase algorithm;

FJ : Fisher and Jaikumar (1981);

TREE : the incomplete tree search algorithm of Christofides, Mingozzi and Toth (1979);

W : Willard's (1989) tabu search algorithm;

PF : the Pureza and França (1991) tabu search algorithm.

The best solution value (when available) obtained with each algorithm is reported in Table 1, together with the computation time (in minutes on a Sun Sparc workstation) for TABUROUTE. Solution values for the other algorithms are given in their respective reference, except for CW, MJ and SWEEP which are taken from Christofides, Mingozzi and Toth (1979). For each problem, the best result is indicated by an asterisk.

These results show that TABUROUTE outperforms all other algorithms on most problems. The comparison may be even more favorable to TABUROUTE than Table 1 indicates. It must first be said that we did not generally possess the full solutions produced by the other algorithms, but only their value. This poses a number of problems. First, it is obvious that rounding or truncating has occurred in the final solution or at some intermediate stage (perhaps even on the distances) since the reported optima are integer while the distances are real. As a result, solution values

reported in Table 1 (other than those obtained with TABUROUTE which did not round or truncate) probably underestimate the true value to some extent. Another difficulty arises in problems with very tight route length constraints, where some routes can be infeasible for TABUROUTE, but feasible when rounding or truncating occurs. We are aware of such cases where TABUROUTE would have achieved a much better value had we considered legal some routes with a length exceeding L by less than one unit. This type of problem has already been reported by Mole (1983) in relation with a vehicle scheduling algorithm by Cheshire, Malleson and Naccache (1982). To eliminate any doubt on our own results and to help other researchers, we produce the full solutions in the Appendix.

Table 1. Comparison of TABUROUTE with ten alternative heuristics

Problem	*n*	*CW*	*MJ*	*AG*	*DV*	*SWEEP*	*TP*	*FJ*	*TREE*	*W*	*PF*	*TABUROUTE*	
												Solution value	*Time*
1	50	585	575	564	586	532	547	524*	534	558	536	524.61	5.6
2	75	900	910	878	885	874	883	857	871	893	842	836.37*	19.4
3	100	886	882	868	889	851	851	833	851	906	851	826.14*	35.3
4	150	1204	1259	1104	1133	1079	1093	1014*	1064	- - -	1081	1034.90	96.1
5	199	1540	1545	1370	1424	1389	1418	1386	1386	- - -	- - -	1329.29*	141.6
6	50	619	599	617	593	560	565	560	560	- - -	560	555.43*	14.1
7	75	976	969	939	963	933	969	916	924	- - -	929	913.23*	46.8
8	100	973	999	938	914	888	915	885	885	- - -	887	865.94*	25.1
9	150	1426	1289	1236	1292	1230	1245	1217	1217	- - -	1227	1189.79*	38.4
10	199	1800	1770	1499	1559	1518	1508	1508	1509	- - -	- - -	1421.88*	83.7
11	120	1079	1100	1066	1058	1266	1066	- - -	1092	- - -	- - -	1043.94*	51.9
12	100	831	879	845	828	937	827	824	816*	- - -	- - -	822.85	13.8
13	120	1634	1590	1573	1562	1770	1612	- - -	1608	- - -	- - -	1551.63*	81.3
14	100	877	883	880	882	949	876	848*	878	- - -	866	866.37	97.1

In addition to these tests, we have carried out a comparison with the improved two-phase algorithm of Toth (1984), who used a subset of the same test problems, but with truncated distances. The comparison is reported in Table 2. The results indicate that in all ten cases where the two algorithms were tested, TABUROUTE yields a lower objective value. Finally, comparisons were also made with the algorithms of Harche and Raghavan (1991) and of Noon, Mittenthal and Pillai (1991) who used rounded distances. The results, reported in Table 3, indicate once more that TABUROUTE produces better results. Note that the value of 521 obtained for the first problem is optimal (Cornuejols and Harche, 1991).

Table 2. TABUROUTE vs. Improved two-phase on problems with truncated distances

Problem	*n*	*Improved two phase algorithm*	*TABUROUTE*
1	50	521	508*
2	75	852	812*
3	100	825	790*
4	150	1026	986*
5	199	1353	1272*
6	50	550	535*
7	75	924	876*
8	100	863	827*
9	150	1156	1110*
10	199	1418	1343*
11	120	- - -	1008
12	100	- - -	799
13	120	- - -	1501
14	100	- - -	836

Table 3. TABUROUTE vs. two alternative algorithms on problems with rounded distances

Problem	*n*	*Harche and Raghavan*	*Noon, Mittenthal and Pillai*	*TABUROUTE*
1	50	521*	521*	521*
2	75	847	845	832*
3	100	825	831	815*
4	150	1 070	1 057	1 024*
5	199	1 397	---	1 316*
6	50	---	---	548
7	75	---	---	907
8	100	---	---	856
9	150	---	---	1 180
10	199	---	---	1 404
11	120	1 045	1 037	1 035*
12	100	---	824*	824*
13	120	---	---	1 529
14	100	---	---	866

4. Conclusion

We have described in this paper a tabu search algorithm for the VRP. Results obtained on a series of benchmark problems indicate clearly that the proposed method outperforms the best existing heuristics. By nature, tabu search is a metaheuristic that must be tailored to the shape of the particular problem at hand. We attribute a large part of the success of TABUROUTE to at least two main implementation devices. One is the fact that we allow infeasible solutions through penalty terms in the objective function, thus reducing the likelihood of local minima. This idea is also used by Hertz (1992) in a course scheduling algorithm. A second important ingredient of our

method is the use of GENI to execute the insertions. Not only does this help produce better tours, but as a result, the solution is periodically perturbed and thus, the risk of being trapped in a local optimum is again reduced. Finally, one major advantage of the proposed algorithm lies in its flexibility. It can be executed from any starting solution (feasible or not); it can also be adapted to contexts where the number of vehicles is fixed or bounded, where vehicles have different characteristics, etc. Also, additional features can easily be handled, such as assigning particular customers to specific vehicles, using several depots, allowing for primary and secondary routes, and so on.

ACKNOWLEDGEMENTS

This work was partially supported by the Canadian Natural Sciences and Engineering Research Council under grants OGP0038816, OGP0039682 and OGP0105384. The second author also benefitted form an NSERC International Fellowship. Thanks are due to Hélène Mercure for her help with programming.

REFERENCES

Altinkemer, K. and B. Gavish, "A Parallel Savings Heuristic for the Delivery Problem with a Log Q Error Guarantee". Graduate School of Management, University of Rochester, NY, 1985.

Bodin, L.D., B.L. Golden, A.A. Assad and M.O. Ball, "Routing and Scheduling of Vehicles and Crews. The State of the Art", *Computers & Operations Research* 10(1983), 69–211.

Cheshire, I.M., A.M. Malleson and P.F. Naccache, "A Dual Heuristic for Vehicle Scheduling", *Journal of the Operational Research Society* 33(1982), 51–61.

Christofides, N., "The Vehicle Routing Problem", *RAIRO (recherche opérationnelle)* 10(1976), 55–70.

Christofides, N., "Vehicle Routing", *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, (E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, Eds.), Wiley, Chichester, 1985a, 431–448.

Christofides, N., "Vehicle Scheduling and Routing", presented at the 12th International Symposium on Mathematical Programming, Cambridge, MA, 1985b.

Christofides, N., A. Mingozzi and P. Toth, "The Vehicle Routing Problem", *Combinatorial Optimization*, (N. Christofides, A. Mingozzi, P. Toth and C. Sandi, Eds.), Wiley, Chichester, 1979, 315–338.

Christofides, N., A. Mingozzi and P. Toth, "State Space Relaxation Procedures for the Computation of Bounds to Routing Problems", *Networks* 11(1981), 145–164.

Clarke, G. and J.W. Wright, "Scheduling of Vehicles from a Central Depot to a Number of Delivery Points", *Operations Research* 12(1964), 568–581.

Cornuejols, G. and F. Harche, "Polyhedral Study of the Capacitated Vehicle Routing Problem". Management Science Research Report #553. Carnegie Mellon University, Pittsburgh, 1991.

Desrochers, M. and T.W. Verhoog, "A Matching Based Savings Algorithm for the Vehicle Routing Problem". Cahier du GERAD G-89–04. École des Hautes Études Commerciales de Montréal, 1989.

Dror, M. and L. Levy, "A Vehicle Routing Improvement Algorithm Comparison of a Greedy and a Matching Implementation for Inventory Routing", *Computers & Operations Research* 13(1986), 33–45.

Fisher, M.L. and R. Jaikumar, "A Generalized Assignment Heuristic for Vehicle Routing", *Networks* 11(1981), 109–124.

Gaskell, T., "Bases for Vehicle Fleet Scheduling", *Operational Research Quarterly* 18(1967), 281–295.

Gendreau, M., A. Hertz and G. Laporte, "New Insertion and Post-Optimization Procedures for the Traveling Salesman Problem". Publication CRT-708. Centre de recherche sur les transports, Montréal, 1990.

Gillett, B. and L. Miller, "A Heuristic Algorithm for the Vehicle Dispatch Problem", *Operations Research* 22(1974), 340–349.

Glover, F., "Heuristic for Integer Programming Using Surrogate Constraints", *Decision Sciences* 8(1977), 156–166.

Glover, F., "Tabu Search, Part I", *ORSA Journal on Computing* 1(1989), 190–206.

Glover, F., "Tabu Search, Part II, *ORSA Journal on Computing* 2(1990), 4–32.

Golden, B.L. and A.A. Assad, "*Vehicle Routing: Methods and Studies*", North-Holland, Amsterdam, 1988.

Golden, B.L. and W.R. Stewart, Jr., "Empirical Analysis of Heuristics", *The Traveling Salesman Problem. A Guided Tour of Combinatorial Optimization*, (E.L. Lawler, J.K. Lenstra, A.H.G. Rinnooy Kan and D.B. Shmoys, Eds.), Wiley, Chichester, 1985, 207–249.

Haimovich, M. and A.H.G. Rinnooy Kan, "Bounds and Heuristics for Capacitated Routing Problems", *Mathematics of Operations Research* 10(1985), 527–542.

Harche, F. and P. Raghavan, "A Generalized Exchange Heuristic for the Capacitated Vehicle Problem". Working paper. Stern School of Business, New York University, 1991.

Hertz, A., "Finding a Feasible Course Schedule Using Tabu Search", *Discrete Applied Mathematics*, 1992. (Forthcoming).

Laporte, G. and Y. Nobert, "Exact Algorithms for the Vehicle Routing Problem", *Surveys in Combinatorial Optimization*, (S. Martello, G. Laporte, M. Minoux and C. Ribeiro, Eds.), North-Holland, Amsterdam, 1984, 147–184.

Lin, S., "Computer Solutions of the Traveling Salesman Problem", *Bell System Computer Journal* 44(1965), 2245–2269.

Lin, S. and B.W. Kernighan, "An Effective Heuristic Algorithm for the Traveling-Salesman Problem", *Operations Research* 21(1973), 498–516.

Mole, R.H., "The Curse of Unintended Rounding Error : a Case from the Vehicle Scheduling Literature", *Journal of the Operational Research Society* 34(1983), 607–613.

Mole, R.H. and S.R. Jameson, "A Sequential Route-Building Algorithm Employing a Generalised Savings Criterion", *Operational Research Quarterly* 27(1976), 503–511.

Nelson, M.D., K.E. Nygard, J.H. Griffin and W.E. Shreve, "Implementation Techniques for the Vehicle Routing Problem", *Computers & Operations Research* 12(1985), 273–283.

Noon, C.E., J. Mittenthal and R. Pillai, "A TSSP+1 Decomposition Approach for the Capacity-Constrained Vehicle Routing Problem". Working paper. Management Science Program, The University of Tennesse, Knoxville, TN, 1991.

Or, I., "Traveling Salesman-Type Combinatorial Optimization Problems and their Relation to the Logistics of Regional Blood Banking", Ph. D. Dissertation, Northwestern University, Evanston, IL, 1976.

Paessens, H., "The Savings Algorithm for the Vehicle Routing Problem", *European Journal of Operational Research* 34(1988), 336–344.

Pureza, V.M. and P.M. França, "Vehicle Routing Problems via Tabu Search Metaheuristic". Publication CRT-747. Centre de recherche sur les transports, Montréal, 1991.

Semet, F. and E. Taillard, "Solving Real-Life Vehicle Routing Problems Efficiently Using Taboo Search". Working paper ORWP 91/03. Département de Mathématiques, École Polytechnique Fédérale de Lausanne, Switzerland, 1991.

Taillard, E., "Robust Taboo Search for the Quadratic Assignment Problem". Working paper ORWP 90/10. Département de Mathématiques, École Polytechnique Fédérale de Lausanne, Switzerland, 1990.

Toth, P., "Heuristic Algorithms for the Vehicle Routing Problem", presented at the Workshop on Routing Problems, Hamburg, 1984.

Willard, J.A.G., "Vehicle Routing Using r-Optimal Tabu Search", M.Sc. Dissertation. The Management School, Imperial College, London, 1989.

Wren, A., *Computers in Transport Planning and Operation*, Ian Allan, London, 1971.

Wren, A. and A. Holliday, "Computer Scheduling of Vehicles from one or more Depots to a Number of Delivery Points", *Operational Research Quarterly* 23(1972), 333–344.

Yellow, P., "A Computational Modification to the Savings Method of Vehicle Scheduling", *Operational Research Quarterly* 21(1970), 281–283.

APPENDIX : Test Problem Solutions

This appendix contains the "best" solutions produced by TABUROUTE for the 14 test problems, using real, rounded, or truncated distances. The tables give the *distance* travelled on each vehicle route. In time constrained problems, route lengths can be obtained by adding to this value the number of clients multiplied by δ. When a table is produced for a given distance type, only the solutions corresponding to that type are the best. For example, in "Problem 2 : best solution with real or rounded distances", the first two columns (with bold type totals) correspond to a best solution. The third column merely computes the value of that solution, using truncated distances. The best solution with truncated distances is given in the next table. In problems with maximum length constraints, some routes may be infeasible: these are shown in italics. For example, in "Problem 6 : best solution with rounded distances", the third route computed with real distances has a length of 111.91 + 90 = 201.91, and is therefore infeasible.

Problem 1 : best solution with real, rounded or truncated distances

number of clients	Route	Q=160 Load	distance real	rounded	truncated
9	0 47 4 17 42 19 40 41 13 18 0	157	109.06	109	104
9	0 6 14 25 24 43 7 23 48 27 0	152	98.45	97	96
10	0 38 9 30 34 50 16 21 29 2 11 0	159	99.33	99	95
11	0 12 37 44 15 45 33 39 10 49 5 46 0	160	99.25	99	98
11	0 8 26 31 28 3 36 35 20 22 1 32 0	149	118.52	117	117
n = 50			524.61	521	508

Problem 2 : best solution with real or rounded distances

number of clients	Route	Q=140 Load	distance real	rounded	truncated
5	0 39 9 32 40 17 0	126	57.01	57	55
6	0 26 58 10 38 65 66 0	135	81.63	81	81
6	0 7 53 11 59 14 35 0	139	84.34	85	82
6	0 67 52 27 45 4 75 0	137	41.22	40	39
7	0 2 28 61 69 21 74 30 0	138	88.71	89	86
8	0 6 33 63 23 56 24 49 16 0	140	92.69	92	90
9	0 73 1 43 41 42 64 22 62 68 0	136	94.63	94	91
9	0 5 37 20 70 60 71 36 47 48 0	135	94.14	93	92
9	0 29 15 57 13 54 19 8 46 34 0	140	82.69	83	79
10	0 12 72 31 25 55 18 50 44 3 51 0	138	119.32	118	118
n = 75			836.37	832	813

Problem 2 : best solution with truncated distances

number of clients	Route	Q=140 Load	distance real	rounded	truncated
5	0 39 9 32 40 17 0	126	57.01	57	55
6	0 26 58 10 65 66 38 0	135	82.76	83	80
6	0 7 53 11 59 14 35 0	139	84.34	85	82
6	0 67 52 27 45 4 75 0	137	41.22	40	39
7	0 2 28 61 69 21 74 30 0	138	88.71	89	86
8	0 6 33 63 23 56 24 49 16 0	140	92.69	92	90
9	0 73 1 43 41 42 64 22 62 68 0	136	94.63	94	91
9	0 5 37 20 70 60 71 36 47 48 0	135	94.14	93	92
9	0 29 15 57 13 54 19 8 46 34 0	140	82.69	83	79
10	0 12 72 31 25 55 18 50 44 3 51 0	138	119.32	118	118
n = 75			837.49	834	812

Problem 3 : best solution with real, rounded or truncated distances

number of clients	Route	Q=200 Load	distance real	rounded	truncated
5	0 94 95 97 87 13 0	108	40.91	40	40
12	0 21 72 75 58 39 67 23 41 22 74 73 40 0	194	106.06	104	102
12	0 28 12 80 68 29 24 54 55 25 4 26 53 0	165	98.25	97	95
13	0 89 60 5 84 45 17 61 85 93 59 99 96 6 0	196	82.73	81	79
14	0 52 7 82 48 19 11 64 49 36 47 46 8 83 18 0	199	138.79	138	132
14	0 50 33 81 51 9 71 65 35 34 78 79 3 77 76 0	199	118.79	119	112
14	0 31 88 62 10 63 90 32 66 20 30 70 1 69 27 0	199	113.93	111	109
16	0 92 98 37 100 91 16 86 38 44 14 42 43 15 57 2 58 0	198	126.66	125	121
n = 100			826.14	815	790

Problem 4 : best solution with real or rounded distances

number of clients	Route	Q=200 Load	distance real	rounded	truncated
4	0 12 63 108 103 0	84	27.72	27	27
10	0 111 66 41 136 13 67 134 55 18 110 0	196	83.96	83	82
11	0 32 51 101 3 59 20 131 83 2 100 11 0	191	74.38	74	69
12	0 47 56 146 149 4 109 143 135 148 147 17 144 0	200	55.14	54	52
12	0 76 49 104 30 105 75 117 89 39 54 10 5 0	195	96.09	95	92
13	0 78 126 16 118 21 79 74 34 50 130 9 62 38 0	187	73.53	72	69
13	0 139 68 133 14 58 25 95 96 24 98 132 6 102 0	199	90.86	90	86
13	0 57 23 69 97 86 43 99 114 61 7 112 48 138 0	196	120.33	120	117
15	0 90 71 123 122 124 125 106 73 33 72 91 45 15 52 37 0	197	89.82	88	85
15	0 77 119 1 120 80 31 82 140 113 26 8 60 81 27 46 0	195	83.56	81	81
15	0 127 53 129 29 128 84 35 85 36 115 121 116 28 70 22 0	195	122.65	123	117
17	0 145 142 87 150 141 19 94 40 88 64 42 92 93 65 107 44 137 0	200	116.85	117	110
n = 150			1034.90	1024	987

Problem 4 : best solution with truncated distances

number of clients	Route	Q=200 Load	distance real	rounded	truncated
4	0 12 63 108 103 0	84	27.72	27	27
10	0 111 66 41 136 13 67 134 55 18 110 0	196	83.96	83	82
11	0 32 51 101 3 59 20 131 83 2 100 11 0	191	74.38	74	69
12	0 47 56 146 149 4 109 143 135 148 147 17 144 0	200	55.14	54	52
12	0 76 49 104 30 105 75 117 89 39 54 10 5 0	195	96.09	95	92
13	0 78 126 16 118 21 79 74 34 50 130 9 62 38 0	187	73.53	72	69
13	0 139 68 133 14 58 25 95 96 24 98 132 6 102 0	199	90.86	90	86
13	0 57 23 69 97 86 43 99 114 61 7 112 48 138 0	196	120.33	120	117
15	0 90 71 123 122 124 125 106 73 33 72 91 45 15 52 37 0	197	89.82	88	85
15	0 77 119 1 120 80 31 82 140 113 26 8 60 81 27 46 0	195	83.56	81	81
15	0 127 53 129 29 128 84 35 85 36 115 121 116 28 70 22 0	195	122.65	123	117
17	0 145 142 87 150 141 19 94 88 40 64 42 92 93 65 107 44 137 0	200	117.26	118	109
150			1035.31	1025	988

Problem 5 : best solution with real or truncated distances

number of clients	Route	Q=200 Load	distance real	rounded	truncated
4	0 127 87 4 60 0	80	26.72	26	25
10	0 61 33 193 194 158 198 53 195 105 96 0	188	41.15	40	39
10	0 192 184 160 115 185 62 116 23 183 104 0	194	84.48	83	81
10	0 171 47 155 36 122 166 20 138 37 88 0	197	99.83	98	98
11	0 76 97 161 110 25 56 118 72 147 181 3 0	192	79.43	79	75
11	0 6 188 16 67 159 182 49 74 144 63 117 0	190	64.15	63	62
11	0 34 175 46 65 176 102 8 178 35 51 26 0	198	60.12	61	55
11	0 100 150 52 170 11 108 69 180 132 7 149 0	197	83.94	84	78
11	0 95 151 55 44 106 32 187 130 40 17 126 0	184	53.00	52	50
13	0 125 45 58 27 83 13 123 128 70 167 179 99 111 0	195	73.37	73	71
13	0 81 12 9 163 148 92 135 146 18 73 145 24 107 0	199	128.09	127	125
13	0 54 30 48 59 103 5 124 154 15 79 153 29 98 0	192	73.59	73	70
13	0 129 119 38 165 164 85 134 84 14 133 177 19 78 0	199	119.13	118	116
14	0 152 139 120 172 21 173 174 82 121 140 94 64 28 101 0	199	70.37	70	67
14	0 112 66 196 186 22 141 91 142 114 156 93 86 2 157 0	185	80.52	80	75
15	0 191 1 136 199 42 68 113 137 89 143 90 41 190 43 197 0	197	91.70	92	87
15	0 168 50 169 71 77 10 80 131 31 162 75 189 57 109 39 0	200	99.70	98	98
n = 199			1329.29	1317	1272

Problem 5 : best solution with rounded distances

number of clients	Route	Q=200 Load	distance real	rounded	truncated
4	0 127 87 4 60 0	80	26.72	26	25
10	0 61 33 193 194 158 198 53 195 105 96 0	188	41.15	40	39
10	0 192 184 160 115 185 62 116 23 183 104 0	194	84.48	83	81
10	0 171 47 155 36 122 166 20 138 37 88 0	197	99.83	98	98
10	0 97 161 110 25 56 118 72 147 181 3 0	177	79.28	78	76
11	0 6 188 16 67 159 182 49 74 144 63 117 0	190	64.15	63	62
11	0 34 175 46 65 176 102 8 178 35 51 26 0	198	60.12	61	55
11	0 100 150 52 170 11 108 69 180 132 7 149 0	197	83.94	84	78
12	0 95 151 55 44 106 32 187 130 40 76 17 126 0	199	53.29	52	51
13	0 125 45 58 27 83 13 123 128 70 167 179 99 111 0	195	73.37	73	71
13	0 81 12 9 163 148 92 135 146 18 73 145 24 107 0	199	128.09	127	125
13	0 54 30 48 59 103 5 124 154 15 79 153 29 98 0	192	73.59	73	70
13	0 129 119 38 165 164 85 134 84 14 133 177 19 78 0	199	119.13	118	116
14	0 152 139 120 172 21 173 174 82 121 140 94 64 28 101 0	199	70.37	70	67
14	0 112 66 196 186 22 141 91 142 114 156 93 86 2 157 0	185	80.52	80	75
15	0 191 1 136 199 42 68 113 137 89 143 90 41 190 43 197 0	197	91.70	92	87
15	0 168 50 169 71 77 10 80 131 31 162 75 189 57 109 39 0	200	99.70	98	98
n = 199			1329.41	1318	1274

Problem 6 : best solution with rounded distances

number of clients	Route	Q=160 Load	L=200, δ=10 distance real	rounded	truncated
3	0 18 4 47 0	75	39.62	40	37
8	0 17 42 19 40 41 13 25 14 0	131	109.94	109	107
9	0 48 8 31 26 7 43 24 23 6 0	129	*111.91*	110	109
10	0 46 38 9 30 34 50 21 29 16 11 0	134	91.00	90	87
10	0 5 49 10 39 33 45 15 44 37 12 0	155	99.12	99	96
10	0 27 1 22 28 3 36 35 20 2 32 0	153	*101.35*	100	100
n = 50			-	548	536

Problem 6 : best solution with real distances

number of clients	Route	Q=160 Load	L=200, δ=10 distance real	rounded	truncated
4	0 18 4 47 46 0	80	42.33	43	39
8	0 17 42 19 40 41 13 25 14 0	131	109.94	109	107
9	0 27 48 8 26 7 43 24 23 6 0	133	100.64	99	98
9	0 1 22 31 28 3 36 35 20 2 0	137	108.08	107	106
10	0 38 9 30 34 50 21 29 16 11 32 0	141	95.33	94	91
10	0 5 49 10 39 33 45 15 44 37 12 0	155	99.12	99	96
n = 50			555.43	551	537

Problem 6 : best solution with truncated distances

number of clients	Route	Q=160 Load	L=200, δ=10 distance real	rounded	truncated
3	0 18 4 47 0	75	39.62	40	37
9	0 12 17 42 19 40 41 13 25 14 0	160	*110.59*	110	107
9	0 27 48 8 26 7 43 24 23 6 0	133	100.64	99	98
9	0 1 22 31 28 3 36 35 20 2 0	137	108.08	107	106
10	0 38 9 30 34 50 21 29 16 11 32 0	141	95.33	94	91
10	0 46 5 49 10 39 33 45 15 44 37 0	131	99.25	99	96
n = 50			-	549	536

Problem 7: best solution with truncated distances

number of clients	Route	Q=140 Load	L=160, δ=10 distance real	rounded	truncated
5	0 53 11 66 65 38 0	129	77.16	77	75
6	0 75 4 27 52 34 67 0	135	37.80	38	36
7	0 51 83 23 56 41 43 33 0	115	85.24	85	82
7	0 12 72 39 31 10 58 26 0	123	81.69	81	81
7	0 3 44 55 25 9 40 17 0	131	91.38	92	88
7	0 73 1 42 64 22 62 2 0	119	90.34	91	87
7	0 6 16 49 24 18 50 32 0	133	86.85	86	85
7	0 20 70 60 71 36 47 48 0	100	92.75	92	90
7	0 8 54 19 59 14 35 7 0	127	94.62	98	90
7	0 30 74 21 69 61 28 68 0	122	88.64	88	86
8	0 45 29 5 37 15 57 13 46 0	130	78.62	78	76
n = 75			-	-	878

Problem 7 : best solution with rounded distances

number of clients	Route	Q=140 Load	L=160, δ=10 distance real	rounded	truncated
6	0 32 50 18 55 25 9 0	113	92.97	93	91
6	0 37 20 70 60 71 36 0	75	93.85	93	93
6	0 7 14 59 19 8 67 0	131	79.99	80	78
6	0 38 65 66 11 53 35 0	139	79.51	79	78
7	0 75 73 1 42 64 22 62 0	113	90.42	90	88
7	0 4 27 13 54 52 46 34 0	140	62.58	62	61
7	0 74 21 69 61 28 2 68 0	126	87.68	87	86
7	0 63 23 56 41 43 33 6 0	122	83.85	83	81
7	0 26 58 10 31 39 72 17 0	127	84.79	84	84
8	0 30 48 47 5 15 57 29 45 0	138	78.74	77	77
8	0 51 16 49 24 3 44 40 12 0	140	79.42	79	78
n = 75			-	907	893

Problem 7 : best solution with real distances

number of clients	Route	Q=140 Load	L=160, δ=10 distance real	rounded	truncated
6	0 32 50 18 55 25 9 0	113	92.97	93	91
6	0 37 20 70 60 71 36 0	75	93.85	93	93
6	0 7 14 59 19 8 67 0	131	79.99	80	78
6	0 38 65 66 11 53 35 0	139	79.51	79	78
7	0 73 1 42 64 22 62 68 0	103	89.84	90	87
7	0 4 27 13 54 52 46 34 0	140	62.58	62	61
7	0 75 74 21 69 61 28 2 0	136	87.68	88	86
7	0 63 23 56 41 43 33 6 0	122	83.85	83	81
7	0 26 58 10 31 39 72 17 0	127	84.79	84	84
8	0 30 48 47 5 15 57 29 45 0	138	78.74	77	77
8	0 51 16 49 24 3 44 40 12 0	140	79.42	79	78
n = 75			913.23	908	892

Problem 8: best solution with truncated distances

number of clients	Route	Q=200 Load	L=230, δ=10 distance real	rounded	truncated
9	0 26 4 56 23 67 39 25 55 54 0	153	107.08	106	103
9	0 89 60 83 8 46 45 17 84 5 0	90	88.60	88	86
11	0 52 7 19 11 64 49 36 47 48 82 18 0	178	117.55	117	112
11	0 1 51 20 66 65 71 35 9 81 33 50 0	163	117.93	118	112
11	0 31 88 62 10 63 90 32 30 70 69 27 0	155	90.12	87	87
12	0 13 87 42 43 14 44 38 86 16 61 96 6 0	196	*111.53*	110	107
12	0 12 80 68 24 29 34 78 79 3 77 78 28 0	169	90.26	90	86
12	0 95 97 92 98 37 100 91 85 93 99 59 94 0	197	60.52	60	55
13	0 58 2 57 15 41 22 75 74 72 73 21 40 53 0	157	83.10	82	79
n = 100			-	858	827

Problem 8 : best solution with real or rounded distances

number of clients	Route	Q=200 Load	L=230, δ=10 distance real	rounded	truncated
9	0 26 4 56 23 67 39 25 55 54 0	153	107.08	106	103
10	0 89 60 83 8 46 45 17 84 5 6 0	93	89.16	88	87
11	0 52 7 19 11 64 49 36 47 48 82 18 0	178	117.55	117	112
11	0 1 51 20 66 65 71 35 9 81 33 50 0	163	117.93	118	112
11	0 31 88 62 10 63 90 32 30 70 69 27 0	155	90.12	87	87
11	0 13 87 42 43 14 44 38 86 16 61 96 0	191	111.40	110	107
12	0 12 80 68 24 29 34 78 79 3 77 78 28 0	169	90.26	90	86
12	0 94 95 97 92 98 37 100 91 85 93 59 96 0	199	59.35	58	56
13	0 58 2 57 15 41 22 75 74 72 73 21 40 53 0	157	83.10	82	79
n = 100			865.94	856	829

Problem 9 : best solution with truncated distances

number of clients	Route	Q=200 Load	L=200, δ=10 distance real	rounded	truncated
9	0 10 54 105 75 39 89 117 73 106 0	137	106.89	107	103
9	0 101 3 121 85 36 115 116 28 70 0	123	*115.35*	*116*	110
10	0 78 38 62 9 130 50 118 16 126 11 0	160	50.21	49	47
10	0 98 97 86 43 99 7 69 23 57 6 0	181	99.05	98	96
10	0 12 144 145 142 87 148 109 4 146 47 0	169	47.43	47	43
10	0 27 138 48 112 61 114 113 26 60 81 0	141	96.46	96	93
11	0 46 102 132 24 96 95 25 58 14 133 68 0	138	87.61	88	82
11	0 150 19 64 40 88 94 41 141 135 143 149 0	187	*92.16*	*92*	87
11	0 37 137 44 107 65 93 42 92 147 17 63 0	129	78.33	79	73
11	0 129 29 128 84 35 59 20 131 83 2 100 0	187	87.64	89	82
11	0 139 110 18 55 134 87 13 136 66 111 56 0	187	81.23	80	79
12	0 32 51 22 80 31 82 140 8 120 1 119 77 0	145	78.04	78	75
12	0 127 53 21 79 74 34 30 104 49 78 5 103 0	180	77.92	78	73
13	0 108 52 15 45 91 72 33 125 124 122 123 71 90 0	171	*70.22*	69	67
n = 150			-	-	1110

Problem 9 : best solution with rounded distances

number of clients	Route	Q=200 Load	L=200, δ=10 distance real	rounded	truncated
8	0 3 121 85 36 115 116 28 70 0	113	115.33	116	111
9	0 10 54 105 75 39 89 117 73 106 0	137	106.89	107	103
10	0 132 98 97 86 43 99 7 69 23 57 0	182	99.57	98	97
10	0 27 138 48 112 61 114 113 26 60 81 0	141	96.46	96	93
10	0 150 19 64 88 40 94 41 141 135 143 0	178	91.69	91	89
10	0 37 44 107 65 93 42 92 137 17 63 0	117	76.77	76	73
10	0 139 110 18 55 134 67 13 136 66 111 0	184	81.23	80	79
11	0 101 22 80 31 82 140 8 120 1 119 77 0	133	88.33	87	85
11	0 46 102 6 24 96 95 25 58 14 133 68 0	137	87.50	87	82
11	0 53 129 29 128 84 35 59 20 131 83 2 0	187	88.05	89	83
12	0 32 51 100 126 16 118 50 130 9 62 38 78 0	176	61.27	59	57
12	0 11 127 21 79 74 34 30 104 49 76 5 103 0	186	77.24	76	74
13	0 12 144 145 147 87 148 142 109 149 4 146 56 47 0	193	51.30	49	49
13	0 108 52 15 45 91 72 33 125 124 122 123 71 90 0	171	70.22	69	67
n = 150			-	1186	1142

Problem 9 : best solution with real distances

number of clients	Route	Q=200 Load	L=200, δ=10 distance real	rounded	truncated
8	0 3 121 85 36 115 116 28 70 0	113	115.33	116	111
9	0 10 54 105 75 39 89 117 73 106 0	137	106.89	107	103
10	0 6 98 97 86 43 99 7 69 23 57 0	181	98.74	97	96
10	0 27 138 48 112 61 114 113 26 60 81 0	141	96.46	96	93
10	0 150 19 64 88 44 94 41 141 135 143 0	178	91.69	91	89
11	0 139 110 18 55 134 67 13 136 66 111 56 0	187	81.23	80	79
11	0 11 101 22 80 31 82 140 8 120 1 119 0	136	88.89	87	86
11	0 46 102 132 24 96 95 25 58 14 133 68 0	138	87.61	88	82
11	0 53 129 29 128 84 35 59 20 131 83 2 0	187	88.05	89	83
11	0 77 32 51 100 16 118 50 130 9 62 38 0	163	60.04	60	56
12	0 103 108 37 44 107 65 93 42 92 137 17 63 0	149	78.86	78	74
12	0 78 126 127 21 79 74 34 104 30 49 76 5 0	182	76.10	75	73
12	0 12 144 145 147 142 87 148 109 149 4 146 47 0	190	49.99	49	46
12	0 52 15 45 91 72 33 125 124 122 123 71 90 0	153	70.11	69	67
n = 150			1189.78	1182	1138

Problem 10 : best solution with rounded distances

number of clients	Route	Q=200 Load	L=200, δ=10 distance real	rounded	truncated
9	0 76 161 110 163 148 92 135 146 18 0	141	*110.14*	110	107
9	0 150 52 11 170 85 164 165 38 119 0	162	109.29	110	105
10	0 60 180 69 108 134 84 14 133 177 19 0	177	95.27	94	91
10	0 189 75 162 31 131 80 10 77 129 71 0	148	97.21	97	95
10	0 23 62 185 89 137 113 68 42 199 136 0	152	97.88	98	95
10	0 103 5 88 37 138 166 20 124 154 15 0	162	94.62	94	92
11	0 95 151 3 181 147 73 145 24 107 63 117 0	157	78.01	78	75
11	0 187 97 9 25 56 118 72 32 106 44 55 0	186	77.08	78	74
11	0 120 171 174 122 36 155 47 59 48 30 54 0	188	76.17	74	74
11	0 193 194 1 191 158 198 53 195 33 61 6 0	199	40.89	40	37
11	0 4 58 83 13 123 128 70 167 65 46 175 0	190	70.83	70	68
12	0 125 98 45 29 79 153 27 179 99 111 87 127 0	189	61.07	59	57
12	0 96 159 67 104 183 116 144 74 49 182 16 188 0	198	70.97	70	69
12	0 152 86 93 156 114 142 91 141 22 186 66 112 0	143	78.61	77	74
12	0 105 192 184 160 115 90 143 41 190 43 197 196 0	199	72.91	72	69
12	0 81 168 12 50 169 57 39 109 130 40 17 126 0	197	60.72	58	57
13	0 34 176 8 102 78 178 35 132 7 51 149 100 26 0	198	63.48	64	57
13	0 157 2 101 28 64 94 140 121 82 173 21 172 139 0	200	65.43	65	63
n = 199			-	1404	1359

Problem 10 : best solution with real distances

number of clients	Route	Q=200 Load	L=200, δ=10 distance real	rounded	truncated
8	0 76 161 163 148 92 135 146 18 0	129	110.10	110	107
9	0 150 52 11 170 85 164 165 38 119 0	162	109.29	110	105
10	0 60 180 69 108 134 84 14 133 177 19 0	177	95.27	94	91
10	0 189 75 162 31 131 80 10 77 129 71 0	148	97.21	97	95
10	0 23 62 185 89 137 113 68 42 199 136 0	152	97.88	98	95
10	0 103 5 88 37 138 166 20 124 154 15 0	162	94.62	94	92
11	0 95 151 3 181 147 73 145 24 107 63 117 0	157	78.01	78	75
11	0 193 194 1 191 158 198 53 195 33 61 6 0	199	40.89	40	37
11	0 4 58 83 13 123 128 70 167 65 46 175 0	190	70.83	70	68
12	0 187 97 9 110 25 56 118 72 32 106 44 55 0	198	78.43	77	75
12	0 120 171 173 174 122 36 155 47 59 48 30 54 0	196	77.56	78	75
12	0 125 98 45 29 79 153 27 179 99 111 87 127 0	189	61.07	59	57
12	0 96 159 67 104 183 116 144 74 49 182 16 188 0	198	70.97	70	69
12	0 152 86 93 156 114 142 91 141 22 186 66 112 0	143	78.61	77	74
12	0 105 192 184 160 115 90 143 41 190 43 197 196 0	199	72.91	72	69
12	0 81 168 12 50 169 57 39 109 130 40 17 126 0	197	60.72	58	57
12	0 157 2 101 28 64 94 140 121 82 21 172 139 0	192	64.03	64	61
13	0 34 176 8 102 78 178 35 132 7 51 149 100 26 0	198	63.48	64	57
n = 199			1421.88	1406	1359

Problem 10 : best solution with truncated distances

number of clients	Route	Q=200 Load	L=200, δ=10 distance real	rounded	truncated
9	0 147 18 146 135 92 148 163 110 9 0	172	*110.64*	*111*	107
9	0 14 133 84 134 85 164 170 11 150 0	158	*111.32*	*112*	107
9	0 45 58 27 179 99 111 4 87 127 0	145	47.88	47	45
10	0 53 198 192 184 160 115 185 137 89 113 0	191	95.24	94	91
10	0 120 171 174 122 166 20 124 154 15 79 0	165	97.88	98	95
10	0 6 61 33 193 194 1 191 195 105 96 0	181	41.35	41	38
11	0 95 151 3 181 73 145 24 107 63 117 188 0	136	77.23	76	74
11	0 60 132 180 69 108 52 38 165 77 119 71 0	198	*95.51*	95	90
11	0 168 81 26 100 50 169 12 130 40 17 126 0	170	56.50	55	54
11	0 109 39 57 189 162 75 31 131 80 10 129 0	181	92.95	*91*	90
11	0 2 86 93 156 114 142 91 141 22 186 112 0	152	77.13	76	73
12	0 76 187 97 161 25 58 118 72 32 106 44 55 0	191	75.39	75	71
12	0 30 48 47 155 36 138 37 88 5 103 59 125 0	200	74.76	73	72
12	0 16 159 182 49 74 144 116 62 23 183 104 87 0	187	79.60	79	77
12	0 34 65 167 70 128 123 13 83 153 29 98 54 0	198	74.81	74	70
13	0 158 197 43 190 41 90 143 68 42 199 135 196 66 0	177	69.43	70	64
13	0 149 51 7 35 177 19 78 178 8 102 176 46 175 0	198	67.98	68	63
13	0 157 101 28 64 94 140 121 82 173 21 172 139 152 0	188	65.43	65	62
n = 199			-	-	1343

Problem 11 : best solution with real or truncated distances

number of clients	Route	Q=200 Load	distance real	rounded	truncated
16	0 8 12 13 14 15 11 10 9 7 6 5 4 3 1 2 88 0	199	134.98	132	131
16	0 40 43 45 48 51 50 49 46 47 44 41 42 39 38 37 95 0	200	199.83	199	197
16	0 105 53 55 58 56 60 63 66 64 62 61 65 59 57 54 52 0	200	214.22	213	207
16	0 107 67 69 70 71 74 75 72 78 80 79 77 68 76 73 108 0	199	144.43	145	139
16	0 120 119 81 112 84 117 113 83 108 118 18 114 90 91 92 87 0	185	71.53	69	68
19	0 102 101 99 104 103 100 116 98 110 115 97 94 96 93 89 85 86 111 82 0	195	70.71	70	64
21	0 109 21 20 23 26 28 32 35 29 36 34 31 33 30 27 24 22 25 19 16 17 0	197	208.46	208	204
n = 120			1043.94	1038	1008

Problem 11 : best solution with rounded distances

number of clients	Route	Q=200 Load	distance real	distance rounded	distance truncated
16	0 8 12 13 14 15 11 10 9 7 6 5 4 3 1 2 88 0	199	134.96	132	131
16	0 40 43 45 48 51 50 49 46 47 44 41 42 39 38 37 95 0	200	199.63	199	197
16	0 105 53 55 58 56 60 83 66 64 62 61 65 59 57 54 52 0	200	214.22	213	207
16	0 106 107 69 70 67 71 74 75 72 78 80 79 77 68 76 73 0	199	145.32	144	141
16	0 120 119 81 112 64 117 113 83 108 116 18 114 90 91 92 87 0	185	71.53	69	66
19	0 102 101 99 104 103 100 116 98 110 115 97 94 96 93 89 85 86 111 82 0	195	70.71	70	64
21	0 109 21 20 23 26 28 32 35 29 36 34 31 33 30 27 24 22 25 19 16 17 0	197	208.46	208	204
n = 120			1044.83	1035	1010

Problem 12 : best solution with real or rounded distances

number of clients	Route	Q=200 Load	distance real	distance rounded	distance truncated
6	0 67 65 63 74 62 66 0	150	43.59	44	42
8	0 13 17 18 19 15 16 14 12 0	190	95.88	96	95
8	0 57 55 54 53 56 58 60 59 0	200	101.88	101	101
9	0 32 33 31 35 37 38 39 36 34 0	200	97.23	97	95
9	0 98 96 95 94 92 93 97 100 99 0	190	95.94	97	93
10	0 91 89 88 85 84 82 83 86 87 90 0	170	76.07	77	72
11	0 20 24 25 27 29 30 28 26 23 22 21 0	170	50.80	51	50
12	0 5 3 7 8 10 11 9 6 4 2 1 75 0	180	59.62	60	56
13	0 47 49 52 50 51 48 45 46 44 40 41 42 43 0	160	64.81	65	63
14	0 69 68 64 61 72 80 79 77 73 70 71 76 78 81 0	200	137.02	136	135
n = 100			822.85	824	802

Problem 12 : best solution with truncated distances

number of clients	Route	Q=200 Load	distance real	distance rounded	distance truncated
6	0 65 63 74 62 66 67 0	150	43.84	44	42
8	0 13 17 18 19 15 16 14 12 0	190	95.88	96	95
8	0 57 55 54 53 56 58 60 59 0	200	101.88	101	101
9	0 32 33 31 35 37 38 39 36 34 0	200	97.23	97	95
9	0 98 96 95 94 92 93 97 100 99 0	190	95.94	97	93
10	0 91 89 88 85 84 82 83 86 87 90 0	170	76.07	77	72
11	0 20 22 24 25 27 29 30 28 26 23 21 0	170	50.84	52	48
12	0 5 3 7 8 10 11 9 6 4 2 1 75 0	180	59.62	60	56
13	0 47 49 52 50 51 48 46 45 44 42 40 41 43 0	160	65.48	66	62
14	0 69 68 64 61 72 80 79 77 73 70 71 76 78 81 0	200	137.02	136	135
n = 100			823.81	826	799

Problem 13 : best solution with truncated distances

number of clients	Route	Q=200 Load	L=720, δ=50 distance real	rounded	truncated
10	0 105 106 107 104 103 116 100 99 101 102 0	86	38.45	37	34
10	0 53 55 58 54 52 40 42 39 38 37 0	141	214.02	213	211
10	0 43 45 48 51 50 49 47 46 44 41 0	116	190.18	189	188
10	0 28 31 27 30 33 34 36 35 32 29 0	61	195.35	193	191
10	0 57 59 65 61 62 64 66 63 60 56 0	126	208.77	209	204
11	0 73 71 74 72 75 78 80 79 77 76 68 0	141	136.49	136	133
11	0 109 21 26 20 23 25 24 22 19 16 17 0	136	*171.58*	*172*	169
12	0 119 81 2 1 3 4 5 6 83 113 117 82 0	138	107.45	108	105
12	0 120 70 69 67 98 110 115 97 94 93 96 95 0	144	118.68	119	116
12	0 88 111 86 85 112 84 18 114 90 91 89 87 0	129	40.10	39	35
12	0 8 12 13 14 15 11 10 9 7 108 118 92 0	157	*120.19*	119	115
n = 120			-	-	1501

Problem 13 : best solution with real distances

number of clients	Route	Q=200 Load	L=720, δ=50 distance real	rounded	truncated
10	0 53 55 58 54 52 40 42 39 38 37 0	141	214.02	213	211
10	0 43 45 48 51 50 49 47 46 44 41 0	116	190.18	189	188
10	0 28 31 27 30 33 34 36 35 32 29 0	61	195.35	193	191
10	0 57 59 65 61 62 64 66 63 60 56 0	126	208.77	209	204
10	0 21 20 26 23 25 24 22 19 16 17 0	123	170.54	170	169
11	0 105 106 107 104 103 116 100 99 101 102 88 0	90	42.72	41	38
11	0 109 8 12 13 14 15 11 10 9 7 108 0	150	124.96	124	121
11	0 73 71 74 72 75 78 80 79 77 76 68 0	141	136.49	136	133
12	0 119 81 2 1 3 4 5 6 83 113 117 82 0	138	107.45	108	105
12	0 120 70 69 67 98 110 115 97 94 93 96 95 0	144	118.68	119	116
13	0 111 86 85 112 84 18 118 114 90 91 89 92 87 0	145	42.50	41	38
n = 120			1551.63	1541	1514

Problem 13: best solution with rounded distances

number of clients	Route	Q=200 Load	L=720, δ=50 distance real	rounded	truncated
10	0 105 106 107 104 103 116 100 99 101 102 0	86	38.45	37	34
10	0 53 55 58 54 52 40 42 39 38 37 0	141	214.02	213	211
10	0 43 45 48 51 50 49 47 46 44 41 0	116	190.18	189	188
10	0 28 31 27 30 33 34 36 35 32 29 0	61	195.35	193	191
10	0 57 59 65 61 62 64 66 63 60 56 0	126	208.77	209	204
11	0 73 71 74 72 75 78 80 79 77 76 68 0	141	136.49	136	133
11	0 21 20 26 23 25 24 22 19 16 17 108 0	135	*170.59*	170	168
12	0 119 81 2 1 3 4 5 6 113 84 112 82 0	138	108.01	108	106
12	0 120 69 70 67 98 110 115 97 94 96 93 95 0	144	118.76	118	117
12	0 88 111 86 85 18 118 114 90 91 89 92 87 0	131	39.36	38	35
12	0 109 8 12 13 14 15 11 10 9 7 83 117 0	158	*122.01*	120	118
n = 120			-	1529	1505

Problem 14 : best solution with truncated distances

number of clients	Route	Q=200 Load	L=1040, δ=90 distance real	rounded	truncated
3	0 42 40 41 0	40	42.82	43	41
8	0 57 59 60 58 56 53 54 55 0	200	101.88	101	101
9	0 13 17 18 19 15 16 14 12 10 0	200	98.04	96	94
9	0 32 33 31 35 37 38 39 36 34 0	200	97.23	97	95
10	0 75 2 4 6 9 11 8 7 3 5 0	160	56.17	56	53
10	0 69 66 68 64 61 72 62 63 65 67 0	150	57.23	57	55
10	0 74 80 79 77 73 70 71 76 78 81 0	200	129.19	129	127
10	0 47 49 52 50 51 48 45 44 46 43 0	120	59.05	59	57
10	0 91 89 88 85 84 82 83 86 87 90 0	170	76.07	77	72
10	0 1 99 100 97 93 92 94 95 96 98 0	200	98.70	98	93
11	0 21 23 26 28 30 29 27 25 24 22 20 0	170	*50.84*	*52*	48
n = 100			-	-	836

Problem 14 : best solution with real or rounded distances

number of clients	Route	Q=200 Load	L=1040, δ=90 distance real	rounded	truncated
5	0 43 42 44 40 41 0	60	45.47	46	44
8	0 57 59 60 58 56 53 54 55 0	200	101.88	101	101
9	0 20 49 52 50 51 48 45 46 47 0	110	61.56	61	61
9	0 13 17 18 19 15 16 14 12 10 0	200	98.04	98	94
9	0 32 33 31 35 37 38 39 36 34 0	200	97.23	97	95
10	0 75 2 4 6 9 11 8 7 3 5 0	160	56.17	56	53
10	0 23 26 28 30 29 27 25 24 22 21 0	160	49.41	50	48
10	0 69 66 68 64 61 72 74 62 65 67 0	150	57.79	57	56
10	0 63 80 79 77 73 70 71 76 78 81 0	200	128.04	127	127
10	0 91 89 88 85 84 82 83 86 87 90 0	170	76.07	77	72
10	0 1 99 100 97 93 92 94 95 96 98 0	200	98.70	98	93
n = 100			866.37	866	844